II. REMARKS

Claims 1, 6, 9, 12 and 13 have been amended. Specifically, claims 1 and 9 have been amended to recite the

> “information processing unit...manages at least one series of data having a stipulated relationship, each series of data being given a space ID and the processor of each memory module manages a table that contains one or more of said space ID, the logical address of a portion of the series of data managed, the size of said portion and the size of the series of data, in which said processor of each memory module manages said portion of series of data and, in response to an instruction including the space ID and the logical address from the CPU module, the processor of each memory module determines if the portion of the series of data managed is involved in the received instruction by reviewing the space ID and the logical address....,”

as supported by original claims 1 and 9, and by Figs. 8A to 8C and 10A to 10C, and on page 12, line 25, to page 13, line 8, and on page 22, line 17, to page 23, line 15, of the application as originally filed.

Claim 6 has been amended to recite “”’wherein connections between each bus and memory module are repeated to form multi-stage connections among memory modules” as shown in Figures 1 and 7 of the application as originally filed.

Claims12 and 13 each have been amended to recite that “the plurality of sets of buses are connected in parallel between memory modules” as supported by Figure 1 of the application as originally filed.

No new matter has been added by the above amendments.

A. The Invention

The presently claimed invention pertains broadly to apparatuses having parallel computing architecture, such as is used to implement a Single Instruction Stream, Multiple Data Stream (“SIMD”) architecture for performing general purpose parallel processing using appropriate and high-speed memory control. More specifically, the present invention pertains to a computer system having architecture of a parallel computer, wherein the computer system has the features recited in claim 1 of the present application. In addition, the present invention also pertains to an information processing unit having the features recited in claim 9 of the present application.

Various other embodiments, in accordance with the present invention, are recited in the dependent claims. All of the various embodiments, in accordance with the present invention, advantageously utilize an architecture of a parallel computer so that parallel processing by means of appropriate and high-speed memory control can be achieved.

B. <u>The Rejections</u>

Claims 12 and 13 stand rejected under 35 U.S.C. § 112, first paragraph, for lacking enablement. Claim 6 stands rejected under 35 U.S.C. § 112, second paragraph, as indefinite.

Claims 1-13 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Thiel (European Document EP 0 408 810 A1) in view of Hennessy (Computer Organization and Design, at 16-81, 541 and 712-713).

Applicant respectfully traverses the rejection and requests reconsideration of the instant claims for the following reasons.

C. Applicant's Arguments

In view of the present amendment, claims 1-13 are now in full compliance with 35 U.S.C. § 112. In particular, claims 12 and 13 have been amended to recite "wherein the plurality of sets of buses are connected in parallel between memory modules" as illustrated in the instant specification on page 12, lines 4-20; page 17, lines 12-21 and lines 30-34; page 20, lines 19-26, and on page 23, lines 16-19. A person of ordinary skill in the art would know how simultaneous transmission of data to/from the plurality of memory modules is accomplished by reviewing Applicant's specification.

D. The Section 103 Rejection

A prima facie case of obviousness requires a showing that the scope and content of the prior art teaches each and every element of the claimed invention, and that the prior art provides some teaching, suggestion or motivation to combine the references to produce the claimed invention. In re Oetiker, 24 U.S.P.Q.2d 1443 (Fed. Cir. 1992); In re Vaeck, 20 U.S.P.Q.2d 1438 (Fed. Cir. 1991).

In the present case, the Examiner's rejection of the instant claims 1-11 under 35 U.S.C. § 103 is clearly untenable because the scope and content of the prior art fails to teach all of the claimed subject matter.

i. The Thiel Document

European Publication EP 0 408 810 A1 to Thiel (hereafter, the "Thiel Document") teaches a multi-processor computer system that includes sixteen data processors (32a,.., 32p), each connected to a single communication bus (36), (See Abstract, and Figure 5). The communication bus (36) includes a data bus (40) for carrying data and an address bus (38) for carrying associated labeling information uniquely identifying the data (See Abstract). Each processor (32) includes read and write detectors (52) connected to the address bus (38) for detecting labeling information for data required by, or presently stored in, the respective processor (32), (See Abstract). A bulk memory (44) having read and write detectors (46) is connected to the communication bus (36) and an address generator (42) supplies labeling addresses to the address bus (38), (See Abstract).

The Thiel Document teaches that, for each address, one processor (32) or the bulk memory (44) supplies the corresponding data to the data bus (40), and other processors and/or the bulk memory requiring the data read the data from the data bus (40), (See Abstract). In this way, data is transferred between processors and/or the bulk memory wherein the address bus (38) and the read and write decoders (52), (46) are configured for <u>multi-dimensional addressing</u>.

<u>No Architecture of a Parallel Computer</u>

A person of ordinary skill in the art would immediately recognize that the Thiel Document teaches a computer system configured for multi-dimensional addressing, but not a "computer system having architecture of a parallel computer" as recited in claim 1. Likewise, the Thiel Document <u>does not teach</u>, or even suggest, "<u>architecture of a parallel computer</u>" manages (i) "at least one series of data having a stipulated relationship, each series of data being given a space ID," and (ii) "the processor of each memory module manages a table that contains one or more sets of said space ID, the logical address of a portion of the series of data managed, the size of said portion and the size of the series of data" as recited in independent claims 1 and 9 in accordance with the present invention. A person of ordinary skill in the art would realize that, in accordance with the present invention, the space ID specifies a series of data having a stipulated relationship, and that the data to be processed, with instruction from the CPU module to the memory modules, is determined not solely by the space ID but also by the logical address. In other words, in accordance with the embodiments recited by claims 1 and 9, the space ID and the logical address are individually introduced and the data to be processed is determined by both the space ID and the logical address.

In accordance with claims 1 and 9 of the present application, then, the data to be processed is specified using one or more sets of the space ID, and the logical address relates to the portion of the series of data managed. The logical address, as recited in claims 1 and 9, corresponds to the logical start address described in the instant specification (e.g., see Figs. 8A to 8C, Figs. 10A to 10C).

Furthermore, according to claims 1 and 9, since the memory module manages one or more sets of the space ID and the logical address relates to the portion of the series of data managed, if the memory module reviews the space ID and determines that the table does not contain any space ID that is consistent with the space ID in the received instruction, the memory module does not perform any processing operation.

These features of the invention affect the operation of the computer system in profound ways. For example, suppose that an array definition is changed in one RAM core of the memory module. The change of array definition corresponds to, for example, insertion of data records and deletion of data records of the series of data. When the change of array definition occurs, each memory module in accordance with the present invention operates as follows. Each memory module receives the instruction indicative of the change of array definition, and the space ID that is related to the series of data whose array definition changes. Then, each memory module determines whether the table contains the space ID that is consistent with the received space ID. If there is no space ID consistent with the received space ID in the table (i.e., no match), the memory module does not continue operations. Instead, only if the memory module finds in the table the space ID consistent with the received space ID (i.e., a match) and the logical address of the data set in the table indicates that the RAM core stores data records downstream of the position of the change of array definition, then, under these circumstances, the memory module updates the logical address of the data set in the table.

In addition, in accordance with the embodiments of the invention recited in claims 1 and 9, the memory module can manage, as stored in the table, a plurality set of the space ID, the logical address of the portion, and the size of the portion. As a non-limiting example, Figs. 8A to 8C of the instant application shows a memory module managing, and storing in the memory, a plurality of space IDs (i.e., space Ids "010," "016," and "212").

The Thiel Document simply does not teach, or suggest, a "computer system having architecture of a parallel computer" as recited by independent claim 1, or an "informational processing unit" that "has architecture of a parallel computer" as recited by independent claim 9. On the contrary, the Thiel Document is limited to teaching a multi-processor computer system that is designed to minimize time consuming data transfers between the distributed local memories of the processors when processing multi-dimensional data arrays. The device of the Thiel Document achieves this objective by using a Multi-Dimensional (MD) Address Generator that is capable of structuring multi-dimensional addresses of data arrays in a compact way for storage in linearly addressed memories.

As shown in Figure 5 of Thiel, the computer system (30) includes a control processor (56), the MD Address Generator (42) and the plurality of processors (32a), (32b), (32p), wherein each processor has a CPU (48a), (48b), (48p), a local memory (50a), (50b), (50p), and a multi-dimensional (MD) Decoder and Address Generator (52a), (52b), (52p). The computer system (30) also includes a control bus (54) connecting the control processor (56) to the MD Address Generator (42) and the plurality of CPUs (48a), (48b), (48p), a MD Parallel Address Bus (38) connecting the MD Address Generator (42) to the plurality of MD Decoders and Memory Address Generators (52a), (52b), (52p) of the processors (32a), (32b), (32p), respectively, and a 64 bit wide data bus (40) for transferring data between local memories (50a), (50b), (50p) and the processors (32a), (32b), (32p), respectively.

As shown in Figure 8 of Thiel, the MD Address Decoder (52) includes N one-dimensional linear decoders (100a), (100b), etc., connected to the MD Address Bus (38). The shift and mask network (104) extracts only certain bits from the MD Address Bus (38) depending on the value held in the value register (103), and shifts the extracted bits down in significance depending on the value held in the shift value registers (106). The output of the shift and mask network (104) is fed to a linear address generator (130) for generating a local memory location address from a decoded MD address.

From the linear address generator (130), the address of a given decoded address is shown in the following form:

Address = BP + (Aa x Da) + (Ab + Db) +.....

where BP is the array base address, Aa, Ab,...are the values of the decoded field addresses for the dimension zero, dimension one, etc., and Da, Db,...are the values stored in the dimension zero field increment register (134a), dimension one field register (134b), etc. (See col. 6, line 55, to col. 7, line 7).

In view of these teachings of the Thiel Document, a person of ordinary skill in the art would recognize many differences between the MD address space and linear addressing described by Thiel and the computer system, in accordance with the presently claimed invention, having architecture of a parallel computer and the processor of each memory module managing one or more sets of the space ID and the logical address portion of the series of data managed.

These differences are evident, as shown in Fig. 5 of Thiel, wherein the MD address is transferred to the MD Decoder and Memory Address Generator, (52a), (52b),...(52p). As shown in Fig. 8 of Thiel, the MD address input is connected to shift and mask network (104), in which the address is shifted and the redundant portion is masked (at 4, col. 5, lines 28-36). The output of the shift and mask network (104) is given to the comparators (110) and (116), and the final output of the linear decoder (100a) indicates that the MD address falls within the range between the lower boundary and the upper boundary (at 4, col. 5, line 37, to col. 6, line 15).

Some of the differences between the teachings of Thiel and the presently claimed invention are easily appreciated by a simple hypothetical. Suppose that the dimension of the MD Address Bus input is set to 1. If the shifted and masked value X (which is, of course, based on the MD address) equals to or is greater than the Xmin (which is the output of the lower boundary register) and equals to or is less than the Xmax (which is the output of the upper boundary register), then the field decode value

(124a) is set to "1," and as a result the MD Decode signal is made "1" also. The decoder (100a) determines whether the MD address falls within the range between Xmin and Xmax, and it determines the MD address. However, if one where to equate the MD address space defined by the range from Xmin to Xmax as the "space ID," such a comparison would be flawed because the MD address space is defined by Xmin and Xmax whereas the "space ID" of the present invention has no such limitation.

In short, a person of ordinary skill in the art would realize from the above description that the Thiel Document does not teach that "the processor of each memory module manages a table that contains one or more sets of said space ID, the logical address of a portion of the series of data managed, the size of said portion and the size of the series of data" as recited in claims 1 and 9.

Assuming, *arguendo*, that the registers (112), (114), (118), (120) of decoders (100a), (100b), etc. and the registers (140), (142) of the address generators (52) may reasonably be construed to define a "table" as recited in claims 1 and 9 of the present application, the content of such a "table" taught by Thiel would be completely different from the content of the "table" recited in the present claims. Specifically, the contents of the registers (112), (114), (118), (120), (140), (142) are upper and lower boundary values of an N+1 dimensional address array, which cannot be reasonably construed to be addresses and size of series of data. Rather, the contents of the registers (112), (114), (118), (120), (140), (142), which pertain to various upper and lower boundaries of an N+1 dimensional address array, plainly do not disclose a "table" having entries for (i) "space ID," (ii) "the logical address of a portion of the series of data managed," (iii) "the size of said portion," and (iv) "the size of the series of data" as recited in claims 1 and 9 of the present application. This is another difference between the device taught by Thiel and the presently claimed invention.

As discussed above, the Thiel Document does not reasonably teach, or even suggest, a "table" that contains a "space ID" in accordance with the present invention. By storing the space ID in the table, it becomes advantageously possible according to the presently claimed invention to generate and delete space Ids depending on the structure of the data that is to be stored or processed (See, for example, Figures 3 and 4 of the present application). This capability eliminates garbage collection and optimizes the use of memory for data series of various dimensionality (See, for example, page 10, line 23, to page 11, line 4, of the present specification).

As discussed above, the Thiel Document does not reasonably teach, or even suggest, a "table" that contains the logical address or size of the portion of the series of data that is managed. As stated previously, the content of the registers (112), (114), (118), (120) of the decoders (100a), (100b), etc. includes numbers that refer to boundaries of address arrays, which are not reasonably interpretable to be logical addresses of data. For the same reason, the Thiel Document does not teach a "table" that contains the "size of the series of data" as recited in claims 1 and 9 of the present application.

Yet another difference between the presently claimed invention and the device taught by the Thiel Document is that Thiel teaches that the output of the adder (138) corresponds to the linear address of local memory location to access the local memory (50), (at 4, col. 6, lines 52-54). Thus, the linear address is calculated based on the output (104), which is also the base of the MD signal as stated above. Moreover, for a device taught in accordance with Thiel, suppose that an array definition is changed in one local memory of the processor (e.g., local memory (50a) of processor (32a) shown in Fig. 5). If the change of array definition occurs in local memory (50a) of the Thiel device, any processors whose local memory stores data records downstream of the position of the change of array definition also have to change their multi-dimensional address space. In other words, for the processors taught by the Thiel Document, the lower boundary register of each dimension and the upper boundary register of each dimension would have to be updated in view of the change in array definition of an upstream processor. This problem, however, does not exist for a computer system made in accordance with claims 1 and 9 of the present application.

No Plurality of Sets of Buses

A person of ordinary skill in the art would also immediately recognize that the Thiel Document does not teach, or even suggest, a "plurality of sets of buses that make connections between the CPU module and memory modules" as recited in claim 1 and 9. Specifically, the Thiel Document teaches a single communication bus (36) that makes communications between the local memory of the processors (32a), (32b), (32p) and the bulk memory (44) as shown in Figure 5. The communication bus (36) is a single bus constructed to include component buses, such as data bus (40) and address bus (38). However, communication bus (36) is not connected to make communications with the control processor (56) or the CPUs of the processors (32a), (32b), (32p) as shown in Figure 5. Therefore, communication bus (36) cannot be reasonably construed to be a "plurality of sets of buses that make connections between the CPU module and memory modules" in accordance with the presently claimed invention.

A "control bus," as shown in Figure 5 of the Thiel Document, is provided to make connections between the control processor (56), the bulk memory (44) and the processors (32a), (32b), (32p), but this is only a single bus. Assuming, *arguendo*, that the communication bus (36) and the control bus (See Fig. 5) taught by the Thiel Document may be construed as a "set of buses" making communications between the control processor (56) and the local memories of the processors (32a), (32b), (32p), then Thiel still plainly fails to teach, or even suggest, a "plurality of sets of buses that make connections between the CPU module and memory modules" as recited in claims 1 and 9.

A person of ordinary skill in the art of parallel computing processing would know that bus communication between processors requires the following functionalities: (a) addressing a processor, (b) receiving/transferring data from/to the addressed processor, and (c) receiving/sending instructions from/to the processor. These functionalities are essential to bus communication because, for example, a bus cannot select a processor for data transfer without the addressing functionality, or, for example, data cannot be read by an addressed processor without the instruction sending functionality. Therefore, a person of ordinary skill in the art would immediately recognize that the "control bus," the "data bus" and the "address bus" shown in Figure 5 are the three bus elements (i.e., control, data, and address) that form a single bus.

Examiner's Admissions

However, these are not the only deficiencies of the scope and content of the Thiel Document. As admitted by the Examiner, the Thiel Document teaches that the processors (32) have a CPU and a local memory, by not a "RAM core" as recited in claims 1 and 9 (Office Action, dated December 2, 2004, at 7, lines 3-4, and at 13, lines 16-17; and Office Action, dated July 29, 2005, at 6, lines 20-21). The Examiner also admits that the Thiel Document does not teach, or even suggest, that the memory modules receives a synchronization signal for achieving synchronization with the CPU module and other memory modules in accordance with claim 3 (Office Action, dated December 2, 2004, at 8, lines 12-19; and Office Action, dated July 29, 2005, at 8, lines 3-10).

Additional Comments Regarding the Thiel Document

Applicant reemphasizes that, according to independent claims 1 and 9 of the present application, the data to be processes is specified, in each memory module, by referring to both the space ID and the logical address, and that the memory module manages, as stored in a table, a plurality of sets of the space ID, the logical address of the portion and the size of the portion. The Thiel Document simply does not teach these claimed features. To the contrary, the Thiel Document teaches that the data to be processed is specified only by use of the MD address. The MD Decoder and Memory Address Generator (52) specifies the data to be processed only by referring to the MD address. More specifically, the Thiel Document teaches a system that specifies the data to be processed by determining whether the shifted and masked MD address (i.e., the output from the shift and mask network (104)) falls within the range of Xmin to Xmax.

In addition, the Thiel Document teaches the application of one processor (e.g., processor (32a)) to manage only one MD address space. If the dimension is considered as "1," each processor can manage only one MD address space (of course, the MD address space is 2 or more dimensional). In view of these facts, it is clear that the computer system, in accordance with claims 1 and 9 of the present invention, wherein the space ID and the logical address are individually introduced and in which each memory module can manage a plurality of space Ids, is substantially different from the device taught by Thiel, in which the MD address is merely introduced and each processor can manage only one MD address space.

Another difference between the device taught by Thiel and the computer system of the present invention is that the linear address structure, taught by the Thiel Document, is employed to access the local memory (50). On the other hand, in accordance with claims 1 and 9 of the present invention, the logical address is not used to access the memory but is used to determine the memory module managing the portion of the series of data. In fact, according to the presently claimed invention, the address for accessing the local memory (i.e., RAM core 34) is stored in the table as the "On-chip physical start address" (See, e.g., Figs. 8b and 10a as originally filed).

Yet another difference between the teachings of the Thiel Document and of the presently claimed invention, is that a change of array definition in a processor of the Thiel device causes the downstream processors, whose local memory stores data records downstream of the position of the changed array definition, to have to be updated. On the contrary, the computer system as recited in independent claims 1 and 9, do not have this disadvantage. Even if a change of array definition occurs, the memory module whose table does not contain the space ID that is consistent with the received space ID (which is included in the instruction from the CPU) does not operate to update the table.

For all of the above reasons, the "space ID" recited in claims 1 and 9 does not correspond to the MD address space taught by the Thiel Document, and the "logical address" recited in claims 1 and 9 does not correspond to the linear address taught by the Thiel Document. In fact, the device taught by the Thiel Document is so very different from the subject matter of the presently claimed invention and it cannot reasonably be construed to be a "computer system having architecture of a parallel computer" in accordance with claim 1 of the present invention.

ii. <u>The Hennessy Reference</u>

The excerpt from "Computer Organization and Design," 2nd Edition, 1998, pp. 16-18, 541 and 712-713, (hereafter, the "Hennessy Reference") teaches that a RAM can be used as the local memory in a memory hierarchy of a computer in place of a sequential access memory, and that processors operating in parallel need to cooperate when operating on shared data, which is achieved by synchronization.

It is evident that the Hennessy Reference is used to establish a narrow premise, and that the Hennessy Reference does not teach, or even suggest, (1) an "<u>architecture of a parallel computer</u>" that manages (i) "at least one series of data having a stipulated relationship, each series of data being given a space ID," and (ii) "the processor of each memory module manages a table that contains one or more sets of said space ID, the logical address of a portion of the series of data managed, the size of said portion and the size of the series of data" as recited in claims 1 and 9, and (2) a "<u>plurality of sets of buses</u> that make connections between the CPU module and memory modules" as recited in claims 1 and 9.

Summary of the Scope and Content of the Prior Art

Neither the Thiel Document, nor the Hennessy Reference, teach or suggest a computing system, or an information processing unit, wherein (1) an "architecture of a parallel computer" manages (i) "at least one series of data having a stipulated relationship, each series of data being given a space ID," and (ii) "the processor of each memory module manages a table that contains one or more sets of said space ID, the logical address of a portion of the series of data managed, the size of said portion and the size of the series of data" as recited in claims 1 and 9, and (2) a "plurality of sets of buses that make connections between the CPU module and memory modules" as recited in claims 1 and 9.

Conclusion

Claims 1-13 are in full compliance with the requirements of 35 U.S.C. § 112. The rejection under 35 U.S.C. § 103(a) is untenable and should be withdrawn because neither the Thiel document nor the Hennessy Reference reasonably teaches, or even suggests, (1) an "architecture of a parallel computer" manages (i) "at least one series of data having a stipulated relationship, each series of data being given a space ID," and (ii) "the processor of each memory module manages a table that contains one or more sets of said space ID, the logical address of a portion of the series of data managed, the size of said portion and the size of the series of data" as recited in claims 1 and 9, and (2) a "plurality of sets of buses that make connections between the CPU module and memory modules" as recited in claims 1 and 9.

For all of the above reasons, claims 1-13 are in condition for allowance and a prompt Notice of Allowance is earnestly solicited.

Questions are welcomed by the below-signed attorney for applicant.

Respectfully submitted,

GRIFFIN & SZIPL, PC

Reg# 47,395

for Joerg-Uwe Szipl
Registration No. 31,799

GRIFFIN & SZIPL, PC
Suite PH-1
2300 Ninth Street, South
Arlington, VA 22204
Telephone: (703) 979-5700
E-mail: gands@szipl.com
Facsimile: (703) 979-7429
Customer No.: 24203